A method includes detecting a security token device that is un-formatted with respect to an enterprise, wherein the security token device comprises a first cryptographic authentication key, and formatting, by a processor, the security token device by replacing the first cryptographic authentication key of the security token device with a second cryptographic authentication key that is specific to a security requirement of the enterprise.
FIG. 1
FIG. 3
SMARTCARD FORMATION WITH AUTHENTICATION

RELATED APPLICATION

[0001] This application is a continuation of U.S. patent application Ser. No. 11/469,480 filed on Aug. 31, 2006, which is hereby incorporated by reference.

FIELD

[0002] This invention relates generally to tokens, more particularly, to methods, apparatus, and systems for fabricating smartcards.

DESCRIPTION OF THE RELATED ART

[0003] Smart cards are storage devices with components to facilitate communication with a reader or coupler. They have file system configurations and the ability to be partitioned into public and private spaces that can be made available or locked. They also have segregated areas for protected information, such as certificates, e-purses, and entire operating systems. In addition to traditional data storage states, such as read-only and read/write, some vendors are working with sub-states best described as “add only” and “update only.”

[0004] Smart cards are a way to increase security especially for enterprise systems. Enterprise system often contain valuable information such as financial data, personnel records, strategies, etc., that may be critical for the entity administrating the enterprise system. Moreover, for at least the reasons described above, smart cards may offer a mechanism to control access to data within the enterprise systems. Accordingly, the reasons to use smart card are plentiful.

[0005] An information technology administrator may be charged with providing these smart cards for an enterprise. The administrator typically searches for a vendor to provide the smart cards and then work with the vendor to receive pre-formatted smart cards. This process may involve a significant resources, e.g., time, man-hours, etc., to accomplish. Another conventional method of obtaining formatted smart cards is for the administrator to purchase a device that formats the smart cards. These devices are expensive and may not be have a high return on investment for a small number of employees. Accordingly, there is a need for a mechanism to format smart cards without incurring a significant cost.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Various features of the embodiments can be more fully appreciated, as the same become better understood with reference to the following detailed description of the embodiments when considered in connection with the accompanying figures, in which:

[0007] FIG. 1 illustrates an exemplary system in accordance with an embodiment;

[0008] FIG. 2 illustrates an exemplary token management system in accordance with another embodiment;

[0009] FIG. 3 illustrates an exemplary flow diagram in accordance with yet another embodiment; and

[0010] FIG. 4 illustrates an exemplary computing platform.

DETAILED DESCRIPTION OF EMBODIMENTS

[0011] Embodiments generally relate to systems, apparatus, and methods for formatting tokens, such as smartcards. More specifically, a factory module in an enterprise security system may be configured to format the tokens. The factory module may be configured to detect the presence of a generic, uncustomized smartcard in a smartcard reader associated with a client. The factory module may then customize the generic smartcard according to the requirements for a specified enterprise using the smartcard reader. Accordingly, a security officer does not need to order customized smartcards from a third party manufacturer.

[0012] For simplicity and illustrative purposes, the principles of the present invention are described by referring mainly to exemplary embodiments thereof. However, one of ordinary skill in the art would readily recognize that the same principles are equally applicable to, and can be implemented in, all types of secure computing systems, and that any such variations do not depart from the true spirit and scope of the present invention. Moreover, in the following detailed description, references are made to the accompanying figures, which illustrate specific embodiments. Electrical, mechanical, logical and structural changes may be made to the embodiments without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense and the scope of the present invention is defined by the appended claims and their equivalents.

[0013] FIG. 1 illustrates an exemplary secure system 100 in accordance with an embodiment. It should be readily apparent to those of ordinary skill in the art that the system 100 depicted in FIG. 1 represents a generalized schematic illustration and that other components may be added or existing components may be removed or modified. Moreover, the system 100 may be implemented using software components, hardware components, or combinations thereof.

[0014] As shown in FIG. 1, the secure system 100 includes a server 105, clients 110 and a local network 115. The server 105 may be a computing machine or platform configured to execute a token management system 120 through a multiple user operating system (not shown) in conjunction with the clients 110. For example, in order to assist in the formatting and customization of a token or smartcard, server 105 may maintain a database having information relating to: a serial number for each token or smartcard; a date that each token or smartcard was formatted and customized; an applet version installed on each token or smartcard; and a secure channel key identifier. The server 105 may be implemented with server platforms as known to those skilled in the art from Intel, Advanced Micro Devices, Hewlett-Packard, Dell, etc.

[0015] The server 105 may interact with the clients over the local network 115. The local network 115 may be a local area network implementing an established network protocol such as Ethernet, token ring, FDDI, etc. The local network 115 provides a communication channel for the server 105 and clients 110 to exchange data and commands.

[0016] The clients 110 may be computing machine or platform configured to execute secure and open applications through the multi-user operating system. The clients 110 may be implemented with personal computers, workstations, thin clients, thick clients, or other similar computing platform. The clients 110 may use operating systems such as Linux, Windows, Macintosh or other available operating system.

[0017] Each client 110 may be configured to interface with a security device 125. The security device 125 may be configured to act as a gatekeeper to the client 110. More particularly, a user may use a security token, such as a smart card, to
access the respective client 110. Each client 110 may have a
security client 130 executing to monitor the security device 125.
[0018] The security client 130 may be configured to manage
the token. More specifically, the security client 130 may
enroll the token, recover keys for the token or reset a personal
identification number for the token. The security client 130
may also be configured to interface with the token manage-
ment system 120 and act as a proxy for application program
data units (APDUs) between the token management system
120 and the token. The security client 130 may be further
configured to display user interfaces as the token manage-
ment system 120 directs, i.e., prompting the user for creden-
tials and/or PIN, displaying token status.
[0019] The token management system 120 comprises sev-
eral modules, as depicted in FIG. 2. FIG. 2 shows an exam-
ple architecture of the token management system 120 in
accordance with another embodiment. It should be readily
apparent to those of ordinary skill in the art that the token
management system 120 depicted in FIG. 2 represents a
generalized schematic illustration and that other components
may be added or existing components may be removed or
modified. Moreover, the token management system 120 may
be implemented using software components, hardware compo-
nents, or combinations thereof.
[0020] As shown in FIG. 2, the token management system
120 includes a token processing system (labeled as TPS in
FIG. 2) 205, a token key service (TKS) module 210, a data
recovery manager (DRM) module 215 and a certificate
authority (CA) module 220. The TPS 205 may be configured
to act as a registration authority. The TPS 205 may direct
the enrollment process. The TPS 205 may also be configured
to act as a gateway between security clients 130 and tokens
and the modules of the token management system 120.
[0021] The TKS module 210 may be configured to main-
tain master keys for the tokens. The TKS module 210 may
also store symmetric keys associated with the token. These
keys may be derived from a single master key combined with
smart card serial number or identification number, i.e., the
CID. The manufacturer of the smart card may store these
symmetric keys onto the token. The manufacturer may also
forward the single master key to the administrator of the token
management system 120, who installs the key into the TKS
module 210.
[0022] The DRM module 215 may be configured to main-
tain a database of encrypted subject’s private keys, which can
be recovered on demand by an appropriate process.
[0023] The CA module 220 may be configured to generate
X.509 certificates in response to received subject public key
information and certificate enrollment requests.
[0024] Returning to FIG. 1, the client 110 may also execute
a factory module 135. The factory module 135 may be con-
figured to interface with the security client 130. In some
embodiments, the factory module 135 may be invoked as a
menu option or a command line prompt. In other embodi-
ments, the factory module 135 may execute in the background
until an unformatted token is detected in the security device
125.
[0025] Once invoked the factory module 135 may gather
the information necessary to format the smart card so that it is
customized to an enterprise. For example, formatting may
comprise installing applets onto the smartcard, creating secu-

rity domains, creating applet instances, creating a data area
that is read when the smartcard is first inserted by a user
(which would then initiate a further personalization or cus-
tomization phase), and replacing “answer to reset” (or
“ATR”) codes with a new code that is allocated by the en-
terprise. Formatting may also comprise replacing the crypto-
graphic authentication keys or encryption keys with new ones
which are specific to an enterprise. Formatting may also
include information such as shared users lists, group assign-
ments, access lists, etc. The factory module 135 may then use
the security device 125 to format and customize the inserted
token in accordance to the gathered format information.
Accordingly, an administrator can purchase generic unfor-
matted smart cards and format in-house without incurring a
large cost for a smart card formatter.
[0026] FIG. 3 illustrates an exemplary flow diagram 300 in
accordance with an embodiment. It should be readily appar-
ent to those of ordinary skill in the art that the flow diagram
300 depicted in FIG. 3 represents a generalized schematic
illustration and that other steps may be added or existing steps
may be removed or modified.
[0027] As shown in FIG. 3, in step 305, the factory module
135 may detect the presence of a token, in step 305. More
particularly, the security client 130 may pass a notification to
the factory module 305 of the presence of the token. The
security client 130 may also pass the status of the token to
the factory module 130, in step 310.
[0028] If the factory module 135 determines that the status
is formatted, in step 315, the factory module 135 may allow
the log-on process continue with the security client 130, in
step 320. Otherwise, if the factory module 135 determines
that the status of the token is unformatted, the factory module
135 may be configured to determine format information for
the token. For example, the factory module 135 may signal
the security client 130 requesting information of the intended
user such as access lists, group access, file access, etc.
[0029] In step 330, the factory module 135 may be config-
ured to format the token using the security device 125. One
the format process is completed, the factory module 135 may
notify the completion of the formatting of the token.
[0030] FIG. 4 illustrates an exemplary block diagram of a
computing platform 400 where an embodiment may be prac-
ticed. The functions of the security client and token manage-
ment system may be implemented in program code and
executed by the computing platform 400. The security client
and token management system may be implemented in
computer languages such as PASCAL, C, C++, JAVA, etc.
[0031] As shown in FIG. 4, the computer system 400
includes one or more processors, such as processor 402 that
provide an execution platform for embodiments of the secu-

rity client and token management system. Commands and
data from the processor 402 are communicated over a com-
unication bus 404. The computer system 400 also includes
a main memory 406, such as a Random Access Memory
(RAM), where the security client and token management
system may be executed during runtime, and a secondary
memory 408. The secondary memory 408 includes, for
example, a hard disk drive 410 and/or a removable storage
drive 412, representing a floppy diskette drive, a magnetic
tape drive, a compact disk drive, etc., where a copy of a
computer program embodiment for the security client and
token management system may be stored. The removable
storage drive 412 reads from and/or writes to a removable
storage unit 414 in a well-known manner. A user interfaces
with the security client and token management system with a
keyboard 416, a mouse 418, and a display 420. A display
adapter 422 interfaces with the communication bus 404 and the display 420. The display adapter also receives display data from the processor 402 and converts the display data into display commands for the display 420.

[0032] Certain embodiments may be performed as a computer program. The computer program may exist in a variety of forms both active and inactive. For example, the computer program can exist as software program(s) comprised of program instructions in source code, object code, executable code or other formats; firmware program(s); or hardware description language (HDL) files. Any of the above can be embodied on a computer readable medium, which include storage devices and signals, in compressed or uncompressed form. Exemplary computer readable storage devices include conventional computer system RAM (random access memory), ROM (read-only memory), EPROM (erasable, programmable ROM), EEPROM (electrically erasable, programmable ROM), and magnetic or optical disks or tapes. Exemplary computer readable signals, whether modulated using a carrier or not, are signals that a computer system hosting or running the present invention can be configured to access, including signals downloaded through the Internet or other networks. Concrete examples of the foregoing include distribution of executable software program(s) of the computer program on a CD-ROM or via Internet download. In a sense, the Internet itself, as an abstract entity, is a computer readable medium. The same is true of computer networks in general.

[0033] While the invention has been described with reference to the exemplary embodiments thereof, those skilled in the art will be able to make various modifications to the described embodiments without departing from the true spirit and scope. The terms and descriptions used herein are set forth by way of illustration only and are not meant as limitations. In particular, although the method has been described by examples, the steps of the method may be performed in a different order than illustrated or simultaneously.

[0034] Those skilled in the art will recognize that these and other variations are possible within the spirit and scope as defined in the following claims and their equivalents.

What is claimed is:

1. A method comprising:
detecting a security token device that is un-formatted with respect to an enterprise, wherein the security token device comprises a first cryptographic authentication key; and
formatting, by a processor, the security token device by replacing the first cryptographic authentication key of the security token device with a second cryptographic authentication key that is specific to a security requirement of the enterprise.

2. The method of claim 1, further comprising:
determining format information for the security token device; and
embedding the format information on the security token device.

3. The method of claim 2, wherein the format information comprises at least one of access lists, group assignments, or shared user lists.

4. The method of claim 1, further comprising receiving a notification of a completion of the formatting of the security token device.

5. The method of claim 1, wherein formatting the security token device comprises:
implementing an applet in the security token device.

6. The method of claim 1, wherein formatting the security token device comprises:
creating a secure domain for the security token device.

7. The method of claim 1, wherein formatting the security token device comprises:
creating a data area in the security token device to embed format information.

8. The method of claim 1, wherein formatting the security token device comprises:
embedding encryption keys associated with the enterprise in the security token device.

9. The method of claim 1, wherein the security token device is coupled to a client device associated with the enterprise.

10. An apparatus comprising:
memory to store instructions; and
a processor, operatively coupled to the memory, to execute the instructions to:
detect a security token device that is un-formatted with respect to an enterprise, wherein the security token device comprises a first cryptographic authentication key; and
format the security token device by replacing the first cryptographic authentication key of the security token device with a second cryptographic authentication key that is specific to a security requirement of the enterprise.

11. The apparatus of claim 10, wherein to format the security token device, the processor is further to determine format information for the security token device and embed the format information on the security token device.

12. The apparatus of claim 11, wherein the format information comprises at least one of access lists, group assignments, or shared user lists.

13. The apparatus of claim 10, wherein the processor is further to receive a notification of a completion of the formatting of the security token device.

14. The apparatus of claim 10, wherein to format the security token, the processor is further to at least one of:
install an applet in the security token device;
create a secure domain for the security token device;
create a data area in the security token device to embed format information; or
embed encryption keys associated with the enterprise in the security token device.

15. A non-transitory computer-readable storage medium comprising computer-executable instructions that, when executed by a processor, cause the processor to:
detect a security token device that is un-formatted with respect to an enterprise, wherein the security token device comprises a first cryptographic authentication key; and
format, by the processor, the security token device by replacing the first cryptographic authentication key of the security token device with a second cryptographic authentication key that is specific to a security requirement of the enterprise.

16. The non-transitory computer-readable storage medium of claim 15, wherein to format the security token device, the processor is further to determine format information for the security token device and embed the format information on the security token device.
17. The non-transitory computer-readable storage medium of claim 16, wherein the format information comprises at least one of access lists, group assignments, or shared user lists.

18. The non-transitory computer-readable storage medium of claim 15, wherein the processor is further to receive a notification of a completion of the formatting of the security token device.

19. The non-transitory computer-readable storage medium of claim 15, wherein the processor is further to at least one of: install an applet in the security token device; create a secure domain for the security token device; create a data area in the security token device to embed format information; or embed encryption keys associated with the enterprise in the security token device.

20. The non-transitory computer-readable storage medium of claim 15, wherein the security token device is coupled to a client device associated with the enterprise.

* * * * *